数理解析研究所講究録
1054巻 1998年 32-39

Some Results for Cluster Traveling Salesperson Problem

Hans L. Bodlaender* 山崎 浩一 (Koichi Yamazaki) †

Abstract

In this paper, we consider a generalizion on the Traveling Salesperson Problem, called the Cluster Traveling Salesperson Problem (abbreviated **CTSP**). In **CTSP**, we are given a graph $G = (V, E)$ and a partition of V: $V_1, \cdots, V_k$ (the 'clusters'); the goal is to find an optimum (i.e. shortest) tour which visits at least one city in each cluster. We will show that **CTSP** with triangle inequality is at least as hard to approximate as Minimum Set Cover. We also give an algorithm that solves k-CSTP in $O(2^{3k} \cdot |V|^3)$ time, where k denotes the number of clusters.

1 Introduction

The Traveling Salesperson Problem (for short **TSP**) is one of the most well studied problems in Computer Science. It is well known that **TSP** is an NP-complete problem and that there is no approximation algorithm with constant performance ratio. When the triangle inequality holds, then TSP can be approximated with a performance ratio 3/2 with the well known algorithm of Christofides [2]. For the case that all distance are either one or two, Papadimitriou and Yannakakis gave a 7/6 approximation algorithm [9]. Recentry, Arora showed that there is a polynomial time approximation scheme for Euclidean TSP [1].

In this paper, we consider a generalizion of **TSP** which is called the Cluster Traveling Salesperson Problem (abbreviated **CTSP**): given a graph $G = (V, E)$ and a partition of V: $V_1, \cdots, V_k$, into *clusters*, find an optimum (i.e. shortest) tour which visit at least one city in each set V_i, $1 \leq i \leq k$. We will show that **CTSP** with triangle inequality is at least as hard to approximate as Minimum Set Cover, and we give an algorithm that solves k-CTSP in $O(2^{3k} \cdot |V|^3)$, where k-CTSP denotes the problem with k the number of clusters. This shows that CTSP is fixed parameter tractable, when the number of clusters is taken as parameter.

2 Definitions, notations, and problems

Let $G = (V, E, c)$ be an edge weighted and undirected graph, where c is the edge weight function. A sequence of vertices of V $W = (v_1, \cdots, v_l)$ is a *walk* if $\{v_i, v_{i+1}\} \in E$

*Department of Computer Science, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, the Netherlands. Email: hansb@cs.ruu.nl

†Department of Computer Science and Information Mathematics, The University of Electro-Communications

for all $1 \leq i \leq l-1$. $P = (v_1, \cdots, v_l)$ is a *path* if P is a walk and $v_i \neq v_j$ for all $1 \leq i \neq j \leq l$. $T = (v_1, \cdots, v_l)$ is a *tour* if T is a walk and $v_1 = v_l$. $C = (v_1, \cdots, v_l)$ is a *cycle* if $(v_1, \cdots, v_{l-1})$ is a path and $v_l = v_1$, $v_l \neq v_i$ for $2 \leq i < l$. For any walk $W = (v_1, \cdots, v_l)$ (hence for any path and tour), the walk cost of W $wc(W)$ is defined by $\sum_{1 \leq i \leq l-1} c(\{v_i, v_{i+1}\})$. For any walk $W = (v_1, \cdots, v_l)$ (hence for any path and tour), $V(W)$ denotes $\bigcup_{1 \leq i \leq l} v_i$. $\mathbf{N}$ is the set of nonnegative integers.

- Cluster Traveling Salesperson Problem (**CTSP**)

Instance:	A undirected edge weighted complete graph $G = (V, E, c)$, where $c : E \mapsto \mathbf{N}$ is an edge weight function, and a partition of V: $\{V_1, \cdots, V_k\}$.
Solution:	A tour T in V such that $V(T) \cup V_i \neq \emptyset$ for each $1 \leq i \leq k$.
Measure:	The walk cost of the tour.

We call this problem **k-CTSP** when the number of the partitions k is fixed. We can also distinguish the variant of CTSP, where we ask for a cycle instead of a tour. We call this variant of the problem **CTSP** *of cycle definition.* To simplify discussions, we will mainly discuss **CTSP** rather than **CTSP** of cycle definition.

- Minimum Set Cover (**MSC**)

Instance:	A finite nonempty collection $\mathcal{C} = \{C_1, \cdots, C_m\}$ where $C_i \neq \emptyset$, $C_i \neq \bigcup_{1 \leq j \leq m} C_j$ for $1 \leq i \leq m$.
Solution:	A subcollection $\mathcal{S} = \{S_1, \cdots, S_s\} \subseteq \mathcal{C}$, such that $\bigcup_{S_i \in \mathcal{S}} S_i = \bigcup_{C_i \in \mathcal{C}} C_i$.
Measure:	The minimum cardinarity $\lvert\mathcal{S}\rvert$.

Definition 1 [3] *An* NP optimization problem *A is a four-tuple $(I, sol, m, type)$ such that*

1. *I is the set of the* instances *of A and it is recognizable in polynomial time.*

2. *Given an instance x of I, $sol(x)$ denotes the set of feasible solutions of x. A polynomial p exists such that, for any $y \in sol(x)$, $|y| \leq p(|x|)$. Moreover, for any x and for any y with $|y| \leq p(|x|)$, it is decidable in polynomial time whether $y \in sol(x)$.*

3. *Given an instance x and a feasible solution y of x, $m(x, y)$ denotes the positive integer* measure *of y (often also called the value of y). The function m is computable in polynomial time and is also called the* objective *function.*

4. *$type \in \{max, min\}$.*

The **opt** *denotes the function mapping an instace x to the measure of an optimum solution. Given an instance x and a feasible solution y of x, we define the* performance ratio *of y with respect to x as*

$$R(x,y) = max\left\{\frac{m(x,y)}{\mathbf{opt}(x)}, \frac{\mathbf{opt}(x)}{m(x,y)}\right\}.$$

Definition 2 [3] *Let A and B be two NPO problems. A is said to be* E-reducible *to B, (denoted by $A \leq_E B$), if two functions f and g and a positive constant α exist such that:*

1. *For any $x \in I_A$, $f(x) \in I_B$ is computable in polynomial time.*
2. *For any $x \in I_A$ and for any $y \in sol_B(f(x))$, $g(x,y) \in sol_A(x)$ is polynomial time.*
3. *For any $x \in I_A$ and for any $y \in sol_B(f(x))$, $R_A(x, g(x,y)) \leq 1+\alpha(R_B(f(x),y)-1)$.*

The triple (f, g, α) is said to be an E-reduction *from A to B.*

3 Nearest Neighbor Algorithm for CTSP

There exist many heuristics for the standard TSP problem, that perform quite well in practice. One well known heuristic is the **Nearest Neighbor** algorithm. Assuming that the triangle inequality and symmetry hold, the performance ratio of this heuristic is $O(\log n)$ [10]. It has the following form:

1. Start a path with choosing an arbitary vertex.
2. Suppose w was the last vertex added to the path. Find the vertex v that is closest to w among all vertices not yet on the path. Add v to the path with the edge $\{v, w\}$.
3. When all nodes have been added to the path, add an edge connecting the starting node and the last node added.

We can extend the Nearest Neighbor algorithm to work as a heuristic for CTSP with triangle inequality in the following way. Let $g : V \mapsto I$ be the function such that $g(v) = i$ iff $v \in V_i$. A cluster V_i is *visited* in a tour T if $V(T) \cap V_i \neq \emptyset$ otherwise V_i is *unvisited.* The Nearest neighbor algorithm for **CTSP** with triangle inequality is as follows:

Procedure Nearest neighbor algorithm for **CTSP** with triangle inequality
input: an edge weighted complete graph $G = (V, E, c)$ and a partition of V: $\{V_1, \cdots, V_k\}$.
output: a tour

```
for each v ∈ V do
    TOUR_v := (v);
    while there exists unvisited cluster in TOUR_v do
```

select a nearest vertex u to the last vertex in $TOUR_v$
so that u is in an unvisited cluster in $TOUR_v$;
add u at the end of $TOUR_v$;
od
add v at the end of $TOUR_v$;
od
output minimum cost $TOUR_v$ over all $v \in V$;
end.

Unfortunately, there are cases where the Nearest Neighbor algorithm for **CTSP** gives a very bad performance. Consider the output of the nearest neighbor algorithm for the following graph: $G = (V, E, c)$: $V = \{u_1, \cdots, u_m\} \cup \{v_1, \cdots, v_m\} \cup \{w_1, \cdots, w_m\} \cup \{x_1, \cdots, x_m\}$, $E = \{\{y, z\} \mid y \neq z \in V\}$, and $c(v_i, v_j) = c(w_i, w_j) = 1$, $c(u_i, u_j) = c(x_i, x_j) = D$, for $1 \leq i \neq j \leq m$, $c(u_i, v_j) = c(w_i, x_j) = D + 1$, $c(v_i, w_j) = c(u_i, x_j) = c(v_i, x_j) = c(u_i, w_j) = D+2$ for $1 \leq i, j \leq m$. The partition is $\{V_1 = \{v_1, x_1\}, \cdots, V_{2m} = \{u_m, w_m\}\}$ (see Fig. 1). Obviously, the triangle inequality is satisfied. The cost of the output of the Nearest Neighbor algorithm is $(m+1)(D+1)$: a tour of the form $(v_1, v_2, \cdots, v_m, u_1, u_2, \cdots, u_m, v_1)$ will be given by the algorithm: However the optimum value is $2(m + D + 1)$: consider a tour of the form $(v_1, v_2, \cdots, v_m, w_1, w_2, \cdots, w_m, v_1)$. If we take $D + 1 = m$, then the performace ratio is $\dfrac{m+1}{4} > \dfrac{|V|}{16}$.

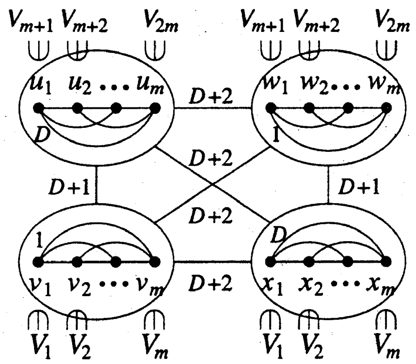


図 1: An example for which the Nearest Neighbor algorithm has bad performance ratio.

4 Hardness of approximating CTSP

In this section we will show that **CTSP** is at least as hard to approximate as **MSC**. It is well known that **MSC** cannot be approximated with ratio $c \log n$ for any $c < 1/4$ unless NP is contained in DTIME($n^{\mathrm{poly}\log n}$) [8]. Hence, under that assumption, there is no approximation algorithm for **CTSP** which has a constant performance ratio. The technique we use in the next theorem is essentially the same one as used in [7].

Theorem 4.1 Minimum Set Cover $\leq_E$ Cluster Traveling Salesperson Problem With Triangle Inequality.

Proof. Let $\mathcal{C} = \{C_1, \cdots, C_m\}$ be an instance for **MSC**. We will construct a complete graph G from the instance $\mathcal{C}$ such that G satisfies the triangle inequality and that the optimum value of **CTSP** for G equals the optimum value of **MSC** for $\mathcal{C}$.

The construction of $G = (V, E, c)$ is as follows: $V = \{(e,i) \mid e \in C_i, 1 \le i \le m\}$, $E = \{\{(e,i),(e',i')\} \mid (e,i),(e',i') \in V, e \ne e' \text{ or } i \ne i'\}$, $c(\{(e,i),(e',i)\}) = 0$ for $1 \le i \le m$ and $c(\{(e,i),(e',i')\}) = 1$ for $1 \le i \ne i' \le m$. The partition of V is $\{V_e = \{(e,i) \mid 1 \le i \le m\} \mid e \in \bigcup_{1 \le i \le m} C_i\}$. See Fig. 2.

It is easy to see that G satisfies the triangle inequality and that the optimum value of **CTSP** for G is exactly equal to the optimum value of **MSC** for $\mathcal{C}$.

Let f be the function that maps the instance $\mathcal{C}$ to the graph G as defined above. Let g be the function mapping a feasible solution $tour \in sol(G)$ to a feasible solution $g(\mathcal{C}, tour) \in sol(\mathcal{C})$ in the following way: Let $((e_1,i_1),(e_2,i_2),\cdots,(e_l,i_l),(e_1,i_1))$ be the $tour$. Then, the corresponding solution $g(\mathcal{C}, tour)$ is $\{S_{i_j} \mid 1 \le j \le l\}$. Without loss of generality, we can assume that $m(\mathcal{C}, g(\mathcal{C}, tour)) = m(G, tour)$.

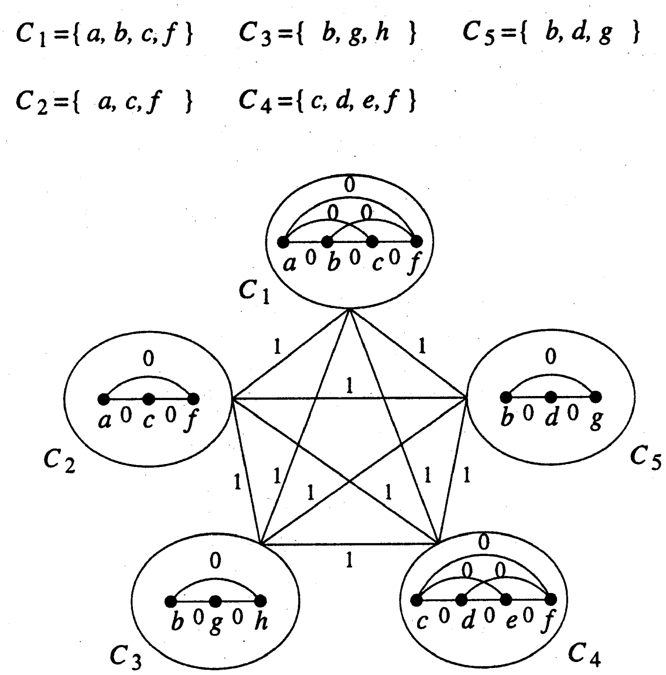


図 2: An example for the construction.

From construction of G, we have $\mathbf{opt}(\mathcal{C}) = \mathbf{opt}(G)$. Thus, we have

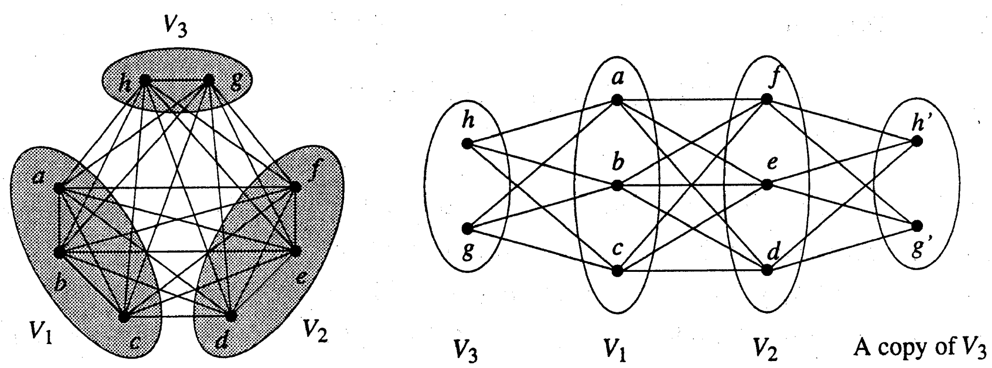
$$\begin{aligned} \frac{m(\mathcal{C}, g(\mathcal{C}, tour))}{\mathbf{opt}(\mathcal{C})} &= \frac{m(G, tour)}{\mathbf{opt}(G)} \\ R_{MSC}(\mathcal{C}, g(\mathcal{C}, tour)) &= 1 + 1(R_{CTSP}(f(\mathcal{C}), tour) - 1). \end{aligned}$$

Thus we have E-reduction $(f, g, 1)$ from **MSC** to **CTSP**. □

5 k-CTSP is Fixed Parameter Tractable

Let k be a fixed integer and G be an edge weighted complete graph of size n with triangle inequality. It is straightforward to solve k-CTSP on graphs with n vertices in $O(n^k \times k!)$ time. However, because the exponent depends on k, such an algorithm would not show that the problem is *fixed parameter tractable*, i.e., is in FPT. In this section, we show that k-CTSP is fixed parameter tractable by giving an algorithm that uses $O(2^{3k} \cdot n^3)$ time. (Recall that the parameter k denotes the number of clusters.) In this section, we will show FTP algorithm for k-**CTSP**.

Let us consider first the situation that we have an oracle that tell us an optimum order in which we should visit the clusters, for a k-**CTSP** with triangle inequality. In this situation, we can compute an optimum tour from the optimum order of clusters in the following way: Let $G = (V, E)$ be the input graph and $(i_1, \cdots, i_k)$ be the order. We construct a new graph $G' = (V', E')$ so that $V' = V \cup V_{i_1}^{copy}$ where $V_{i_1}^{copy}$ is a copy of V_{i_1} and $E' = \{\{p,q\} \in E \mid$ there exists some $j, 1 \leq j < k$ so that $p \in V_{i_j}$ and $q \in V_{i_{j+1}}\} \cup \{\{p,q'\} \mid p \in V_{i_k}$, the vertex $q' \in V_{i_1}^{copy}$ is corresponding to vertex $q \in V_{i_1}$ so that $\{p,q\} \in E\}$. See Fig. 3.



The graph G The new graph G' for order (3,1,2)

図 3: The graphs G and G'.

For each vertex $v \in V_{i_1}$, we denote the corresponding vertex of $V_{i_1}^{copy}$ by v'. For each vertex $v \in V_{i_1}$, we compute a shortest path in G', say $S(v, v')$, between v and v'. Now, choose the shortest path $S_{min}(v, v')$ with minimum cost over all $v \in V_{i_1}$. From the triangle inequality, it follows that $S_{min}(v, v')$ is corresponding to a minimum cost tour in G for k-**CTSP** in which all clusters are visited.

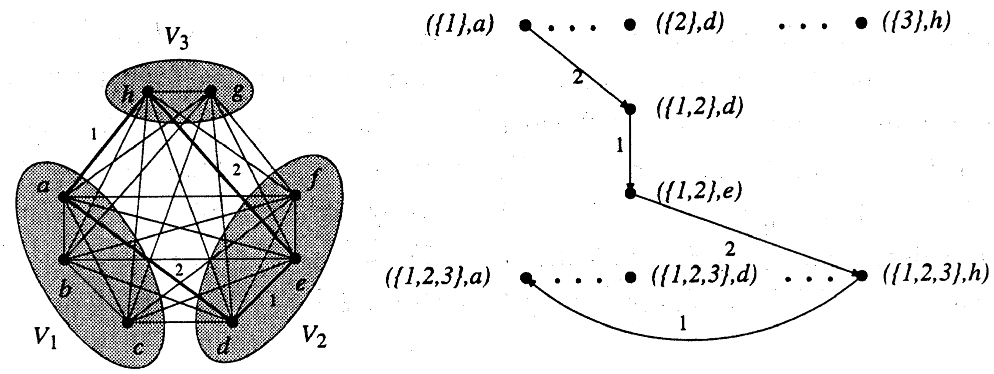
The computation of $S_{min}(v, v')$ for all $v \in V_{i_1}$ can be done in $O(|V'|^3) = O(|V|^3)$ time using an all pairs shortest path algorithm [4]. Since there are $k!$ possible orders for the clusters, k-**CTSP** with triangle inequality can be solved in $O(k!|V|^3)$ time. We also have the following result.

Theorem 5.1 *There exists an* $O(2^{3k} \cdot n^3)$ *time algorithm that solves the* k-**Cluster Traveling Salseperson Problem**, *where* n *is the number of vertices in the input* *graph.*

Proof. Let $G = (V, E, c)$ be an edge weighted complete graph and $V_1, \cdots, V_k$ be a partition of V, $I = \{1, 2, \cdots, k\}$ and $g : V \mapsto I$ such that $g(v) = i$ iff $v \in V_i$.

First, we will construct an edge weighted directed graph $G' = (V', E', c')$ from G:

$$V' = \{(C, u) \mid C \text{ is a nonempty subset of } I, u \in V\},$$
$$E' = \{((C, u), (D, v)) \mid (C, u), (D, v) \in V', D = C \cup \{g(v)\}, \{u, v\} \in E\}, \text{ and}$$
$$c'(((C, u), (D, v))) = c(\{u, v\}).$$



The graph G The graph G'

図 4: The graphs G and G'.

For each $v \in V$, we compute a shortest path between $(\{g(v)\}, v)$ and (I, v), and we denote the shortest path by SP_v. (Notice that $u_1 = r = u_t$). Let $SP_{min} = ((\{g(u_1)\}, u_1), \cdots, (I, u_l))$ be a minimum cost SP_v. Then it is clear that $(u_1, \cdots, u_l)$ is a minimum cost tour in G such that for all $1 \leq i \leq k$, $(u_1, \cdots, u_l)$ visits a city in V_i.

The size of graph G', i.e. $|V'|$, is at most $2^k \times |V|$. Thus, the construction of G' can be done in $O((2^k \times |V|)^2) = O(2^{2k} \cdot |V|^2)$ time. The computation of SP_{min} can be obtained in $O((2^k \cdot |V|)^3) = O(2^{3k} \cdot |V|^3)$ time using for all pairs shortest path algorithm. Therefore, one can solve k-**CTSP** in $O(2^{3k} \cdot |V|^3)$ time. □

For **CTSP** of cycle definition, the method used in Theorem 5.1 does not give a correct answer: the shortest path in G' does not necessarily correspond to a cycle in G. However, if we consider **CTSP** of cycle definition with the triangle inequality, however, then the method of Theorem 5.1 can be used, as any tour can be transformed to a cycle of shorter or equal length that visits the same set of clusters.

6 Open Problems

The following open problems are interesting for further studies.

1. Is there an approximation algorithm for **CTSP** with performance ratio $O(\log n)$ (n the size of the input graph)?

2. Consider the Cluster Traveling Salesperson Problem with edge costs 1 or 2 (for short $CTSP_{1,2}$. It is clear that this problem has an approximation algorithm with ratio 2. How close can we bring the performance ratio of $CTSP_{1,2}$ near to 1?

Acknowledgment

We thank Babette de Fluiter and Dimitrios M. Thilikos for discussions on this research.

参考文献

[1] S. Arora, Polynomial time approximation scheme for euclidean TSP and other geometric problem, Proc. 37th FOCS (1996) 2 – 11.

[2] N. Christofides, Worst-case analysis of a new heuristic for the traveling salesman problem, In Sympos. on New Directions and Recent Results in Algorithms and Complexity, NY Academic Press.

[3] P.Crescenzi, V.Kann, R.Silvestri, and L.Trevisan, Structure in Approximation Classes, manuscript.

[4] T.H.Cormen, C.E.Leiserson, and R.L.Rivest, "Introduction to Algorithms", MIT Press, Cambridge, Mass., USA, 1989.

[5] R. G. Downey and M. R. Fellows, Fixed Parameter Tractability and Completeness I: Basic Results, SIAM J. Computing, 24, (1994) 873 – 921.

[6] A.M.Frieze, G.Galbiati, and F.Maffioli, On the Worst-Case Performance of Some Algorithms for the Asymmetric Traveling Salesman Problem, Networks, 12 (1982) 23 – 39.

[7] E.Ihler, The complexity of approximating the class Steiner tree problem, Proc. Workshop on Graph-Theoretic Concepts in Computer Science, Lecture Notes in Comput. Sci. Springer-Verlag, 570 (1992) 85 – 96.

[8] C.Lund and M.Yannakakis, On the hardness of approximating minimization problems, J. ACM, 41 (1994) 960 – 981.

[9] C. H. Papadimitriou and M. Yannakakis, The traveling salesperson problem with distances one and two, Mathematics of Operations Resarch, 18 (1993) 1 – 11.

[10] D.J.Rosenkrantz, R.E.Stearns, and P.M.Lewis, An analysis of several heuristics for the traveling salesman problem, SIAM J. Comput., 6 (1977) 563 – 581.